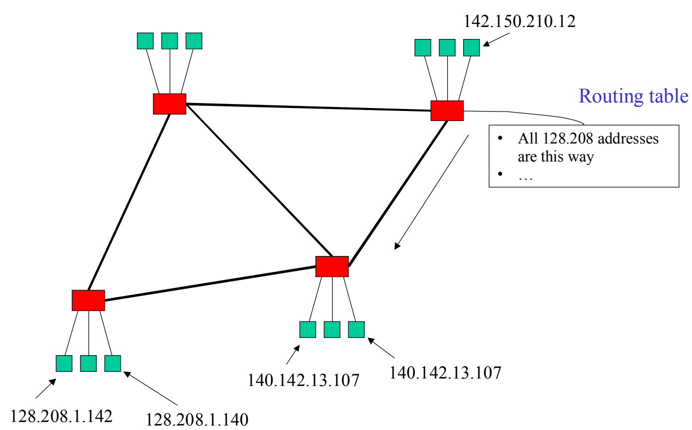
CSE/EE 461: Introduction to Computer Communications Networks
Winter 2009

Module 6
IP Addressing

John Zahorjan
zahorjan@cs.washington.edu
534 Allen Center

Last Time: Addresses Imply Location



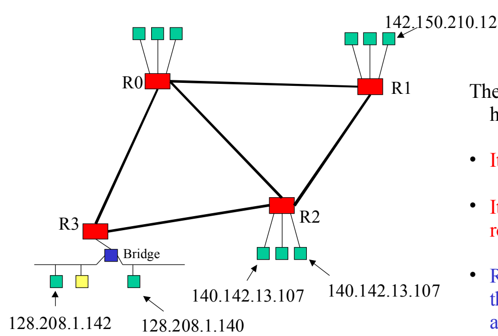
This Lecture

IP Addressing

- Allocation and discovery
  - DHCP
  - ARP
  - NAT
- Hierarchy (prefixes, class A, B, C, subnets)

Application
Presentation
Session
Transport
Network
Data Link
Physical

Address Allocation and Discovery



The yellow node boots. It has a MAC address.

- It needs an IP address.
- It needs to know to use router R3.
- R3 needs to discover the new host's MAC address.

DHCP is used.
ARP is used.

Topic 1: Obtaining an IP address

- Old fashioned way: sysadmin configured each machine
  - E.g., a local file contained the IP address to use
  - Imagine deploying 50 new machines in one of the labs…
- Future fashioned way (IPV6): Stateless Autoconfiguration
  - Addresses are wide / plentiful
  - Form IPv6 address by concatenating “network’s address” (prefix) with your own MAC address
  - Learn “network address” portion from router
- Current (IPv4) way: Dynamic Host Configuration Protocol (DHCP)
  - Addresses are narrow (32-bits) / scarce
    - Have to hand them out carefully
  - Use a DHCP server that provides bootstrap info to hosts
    - Host’s IP address, gateway address, …
  - An immediate problem: how does a host without an IP address communicate with the DHCP server?

The DHCP Problem

- The host doesn’t have an IP address
- The host doesn’t know the address of the DHCP server
- The host wants to contact the DHCP server
- We want to use IP packets to talk with the server
  - Why? Why not talk using link layer packets?

Solution: link and IP layer multicast

The DHCP Problem Solution

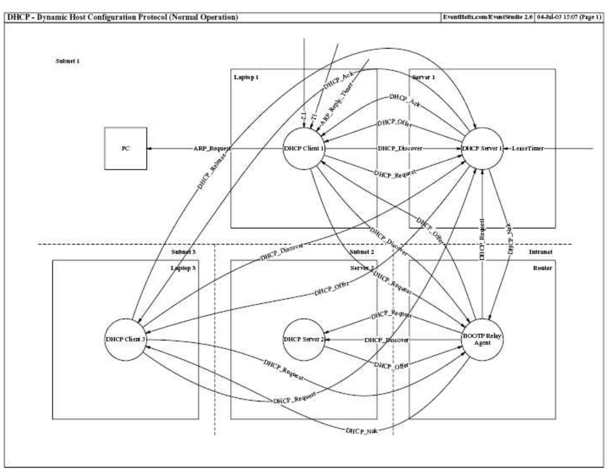
Host

DHCP Server

Ethernet	Src: host's MAC address Dst: FF:FF:FF:FF:FF:FF
IP	Src: 00:00:00:00 Dst: FF:FF:FF:FF
UPD	Src port: 68 Dst port: 67
DHCP	RN32 Hi. Startup info please…

Ethernet	Src: server's MAC address Dst: FF:FF:FF:FF:FF:FF
IP	Src: server's IP address Dst: FF:FF:FF:FF
UPD	Src port: 67 Dst port: 68
DHCP	RN32 Your IP is… Your gateway's IP is… …

(As always, the actual protocol is richer than what is shown here.)

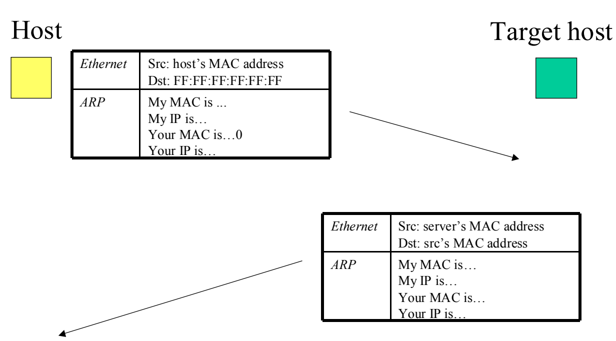


- Client is responsible for all retransmissions. (Why?)
- What dangers are there for losing IP addresses?

Topic 2: Discovering MAC's from IP's

- Host has an IP (e.g., for the gateway). It needs a MAC address to send a frame to it.
- Solution: Address Resolution Protocol (ARP)
- Exploits the physical multicast of Ethernet

The IP->MAC Problem Solution



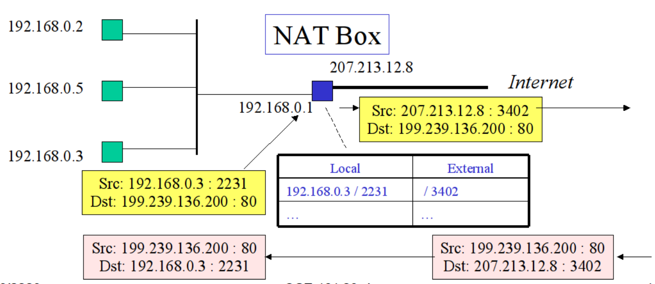
(As always, the actual protocol is richer than what is shown here.)

Topic 3: Network Address Translation (NAT)

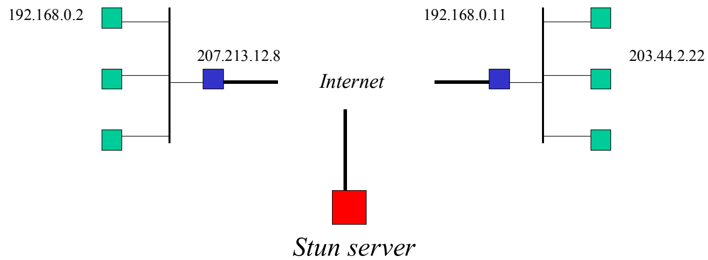
- Turns out that there aren't all that many 32-bit strings (i.e., IP addresses)
  - The world needs more…
  - An individual network needs more…
  - You need more…
    - Your ISP will give you only one (using DHCP), but you want to connect five machines to the Internet
- NAT exploits *non-routable addresses* to let you build your own private network "behind the NAT box"
  - Non-routable addresses are, well, never routed
  - do not have to be globally unique (just locally unique)
- The NAT box substitutes its own IP address for outgoing packets, and the local address of the actual destination for incoming packets

NAT Overview

- Recall that IP addresses are 32-bits (e.g., 192.168.10.3)
- Recall that TCP addresses are IP addresses plus a port number
- These IP address ranges are "non-routable":
  - 10.0.0.0 - 10.255.255.255
  - 172.16.0.0 - 172.31.255.255
  - 192.168.0.0 - 192.168.255.255



NAT and Peer-to-Peer (P2P)



- STUN (Simple Traversal of UDP through NATs)
  - Heuristic designed to discover "routable" address (NAT entry) for hosts behind NATs
  - IETF RFC 3489 (http://tools.ietf.org/html/rfc3489)

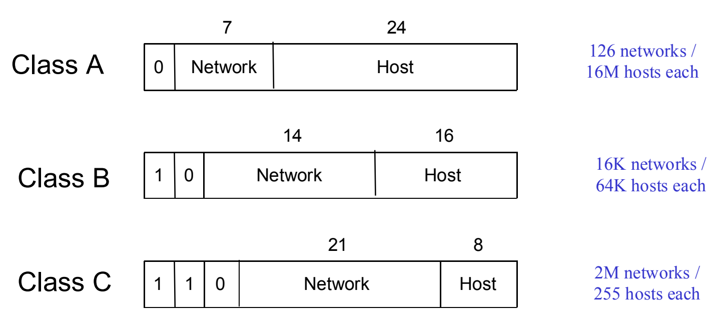
Topic 4: IP Addresses

- Routing burden grows with size of an internetwork
  - Size of routing tables
  - Volume of routing messages
  - Amount of routing computation
- To scale to the size of the Internet, apply:
  - Hierarchical addressing
  - Use of structural hierarchy
  - Route aggregation

IP Addresses: Hierarchy

- Hierarchy is used for <u>routing</u>
  - IP addresses reflect some properties of location in topology
    - Interfaces on the same network share prefix
    - Local delivery in a single network doesn't involve router
    - Routers advertise prefixes to each other
  - Unlike "flat" Ethernet addresses
  - Like hierarchical file names (e.g., `/homes/zahorjan/cse461/09wi/m6.odp`).
    - What are the similarities / differences?
- Hierarchy is used for network <u>management</u>
  - Prefix administratively assigned (IANA or ISP)
    - Addresses globally unique
  - Full host IP assigned locally
    - Distributes burden over users

IPv4 Address Formats



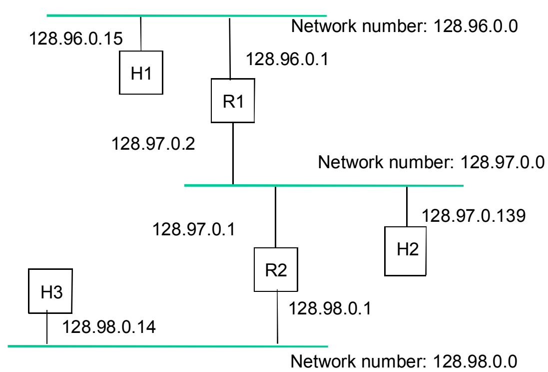
- 32 bits written in "dotted quad" notation, e.g., 18.31.0.135
- Given an IP address, it's easy to determine the network number

IPv6 Address Format

3	5				
001	RegistryID	ProviderID	SubscriberID	SubnetID	InterfaceID

- 128 bits written in 16 bit hexadecimal chunks
- Still hierarchical, just more levels – roughly
  - World owner of this address range (e.g., IANA)
  - Backbone provider
  - ISP
  - End-client organization
  - Interface (host)

IPv4 Network Example

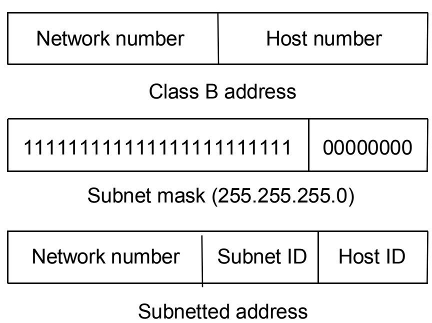


Internet Router Forwarding Routine: Take 1

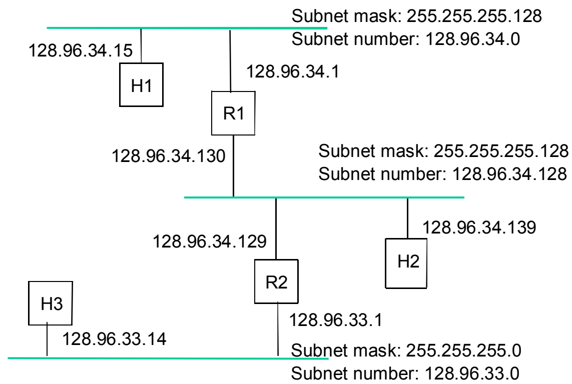
- With bridging, it used to be "look up destination address to determine next hop"
- Now addresses have network and host portions:
  - Host: if destination network is the same as the host network, then deliver locally (without router). Otherwise send to the router
  - Router: look up destination network in routing table to find next hop and send to next router. If destination network is directly attached then deliver locally.

Take 2: Subnetting

- Split up one network number into multiple physical networks
- Helps allocation efficiency -- can hand out subnets
- Rest of internet does not see subnet structure
  - subnet is purely internal to network
  - aggregates routing info



Subnet Example



Updated Forwarding Routine

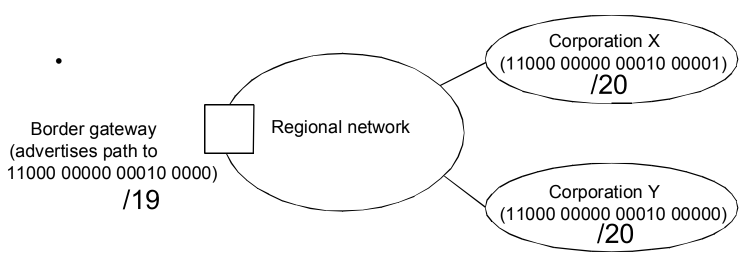
- Used to know network from address (class A, B, C)
- Now need to “search” routing table for right subnet
  - Host: easy, just substitute “subnet” for “network”
  - Router: search routing table for the subnet that the destination belongs to, and use that to forward as before

Take 3: CIDR (Supernetting)

- CIDR = Classless Inter-Domain Routing
- Generalize class A, B, C into prefixes of arbitrary length; now must carry prefix length with address
- Aggregate adjacent advertised network routes
  - e.g., ISP has class C addresses 192.4.16 through 192.4.31
  - Really like one larger 20 bit address class …
  - Advertise as such (network number, prefix length)
  - Reduces size of routing tables
- But IP forwarding is more involved
  - Based on Longest Matching Prefix operation

CIDR Example

- X and Y routes can be aggregated because they form a bigger contiguous range.



- But aggregation isn't always possible
  - can only aggregate power of 2

IP Forwarding Revisited

- Routing table now contains routes to “prefixes”
  - IP address and length indicating what bits are fixed
- Now need to “search” routing table for longest matching prefix, only at routers
  - Search routing table for the prefix that the destination belongs to, and use that to forward as before
  - There can be multiple matches; take the longest prefix
- This is the IP forwarding routine used at routers.

Key Concepts

- Hierarchical address allocation helps routing scale
  - Technical Issues:
    - Addresses are constrained by topology
    - Advertise and compute routes for networks, not each host
    - Separate internet view of networks from local implementation via subnets
    - Keep host simple and let routers worry about routing
  - Network Administration Issue:
    - Distribute workload of assigning IP addresses to clients
- DHCP provides convenient management of host startup information
- ARP learns the mapping from IP to MAC address
- NAT hides local names behind a single global name